

PATENTMETHOD AND APPARATUS OF  
ADDING GRANT INFORMATION TO A MEMORY

## BACKGROUND OF THE INVENTION

5

1. Field of the Invention.

The present invention relates to a method and apparatus of allocating bandwidth on a bus-based communication system and, more particularly, to a method and apparatus of adding bus grant information to a memory that is used to grant access to a bus.

10

2. Description of the Related Art.

15

A bus-based communication system is a system that allows a number of communication circuits to exchange signals with each other over a group of shared electrical pathways. For example, the communication circuits on service cards, such as xDSL and other line cards, can be connected to, and communicate over, a bus.

20

FIG. 1 shows a block diagram that illustrates a conventional bus-based communications system 100. As shown in FIG. 1, system 100 includes a cell bus 110 that has a first bus BUS-A and a pair of first-bus control lines SEL-A0 and SEL-A1. In addition, cell bus 110 also includes a second bus BUS-B, and a pair of second-bus control lines SEL-B0 and SEL-B1. First bus BUS-A utilizes a first group of electrical pathways, such as eight electrical pathways that represent eight bits, while second bus BUS-B utilizes a second group of electrical pathways, such as eight electrical pathways that represent eight bits.

25

As further shown in the FIG. 1 example, system 100 also includes a number of service cards 112 that are connected to cell bus 110. Each

30

service card 112 is also connected to a number of network devices 114 to receive a number of streams of data cells DS. The data cells DS can have different priority levels such that a data cell DS from one network device 114 is preferred over the data cell DS from another network  
5 device 114.

Each service card 112 includes a communication circuit 116 that has a transmit circuit 120 that transmits data cells onto cell bus 110, and a receive circuit 122 that receives data cells from cell bus 110. Communication circuit 116 also includes a logic block 124 that processes  
10 the data cells DS.

In operation, when a number of communication circuits 116 are connected to cell bus 110, one of the communication circuits 116 assumes the role of bus master, while the remaining communication circuits 116 assume the roles of bus slaves and communicate over the  
15 bus as allowed by the bus master.

The bus master controls the timing of the bus along with access to the bus. For example, the bus master can define transmission periods on bus BUS-A and bus BUS-B, and determine the communication circuit 116 that has permission to use each of the transmission periods  
20 on the buses.

FIG. 2 shows a state diagram that illustrates a prior art state machine 200 operating as a bus master. State machine 200 is executed by the logic block 124 of the communication circuit 116 that is the bus master. As shown in FIG. 2, state machine 200 begins at state 210 by  
25 determining whether any requests to use bus BUS-B were received during a first request period.

Requests to use bus BUS-B are received during request periods where each communication circuit 116 that wishes to transmit a data cell over bus 110 outputs a request. The requests are output over  
30 select lines SEL-A0, SEL-A1, SEL-B0, and SEL-B1 to the bus master.

Each request period can be, for example, 12 clock cycles long to support 24 communication circuits 116.

For example, a 1<sup>st</sup> communication circuit can request bus BUS-B during a first-request clock cycle of a request period on select line SEL-A0 and SEL-A1, while a 13th communication circuit can request bus  
5 BUS-B during the first-request clock cycle on select lines SEL-B0 and SEL-B1. In addition, a 2<sup>nd</sup> communication circuit can request bus BUS-B during a second-request clock cycle on select line SEL-A0 and SEL-A1, while a 14th communication circuit can request bus BUS-B during the  
10 second-request clock cycle on select lines SEL-B0 and SEL-B1. Thus, in this example, after 12 clock cycles, each of 24 communications circuits has had a one clock cycle opportunity to request control of bus BUS-B over select lines SEL-A0, SEL-A1, SEL-B0, and SEL-B1.

Each communication circuit 116 outputs a logic value onto the  
15 select lines SEL during its assigned clock cycle to indicate whether a request is being made and, if so, the priority level of the request. For example, a logic value of 0-0 can represent a high priority level, while a logic value of 0-1 can represent a medium priority level. In addition, a logic value of 1-0 can represent a low priority level, while a logic value  
20 of 1-1 can represent no request.

When requests are received during the first request period, state machine 200 moves to state 212 to define a group of requesting circuits that include the communication circuits 116 that submitted a bus control request during the first request period. For example, state machine 200  
25 can define a group that includes only the communication circuits 116 that requested control of bus BUS-B during the first request period.

Following this, state machine 200 moves to state 214 to grant access to one of the group of requesting communication circuits to transmit in the next transmission period on bus BUS-B. Access is  
30 granted by outputting a grant to the requesting communication circuit

116 over the control lines SEL-B0 and SEL-B1. States 212 and 214 can be, for example, eight clock periods long. In addition, an error correction code can be transmitted at the same time on the select lines SEL-A0 and SEL-A1 that are not carrying the grant. Once the grant has  
5 been output, state machine 200 moves to state 216 to wait for a predefined period of time. The total time required to complete states 210-216 can be, for example, 26 clock cycles.

After the predefined time has expired, state machine 200 moves to state 218 to determine whether any requests to use bus BUS-A were  
10 received during a second request period. In the present example, one clock cycle before state machine 200 moves to state 218, the communication circuit 116 that received control over bus BUS-B, begins transmitting a data cell on bus BUS-B.

As with bus BUS-B, requests to use bus BUS-A are also received  
15 during a request period where each communication circuit 116 that wishes to transmit a data cell over bus 110 can output a request. The requests are again output over select lines SEL-A0, SEL-A1, SEL-B0, and SEL-B1 to the bus master, and the request period can also be 12 clock cycles long to support 24 communication circuits 116.

20 For example, a 1<sup>st</sup> communication circuit can request bus BUS-A during a first-request clock cycle of a next request period on select lines SEL-B0 and SEL-B1, while a 13th communication circuit can request bus BUS-A during the first-request clock cycle on select lines SEL-A0 and SEL-A1. In addition, a 2<sup>nd</sup> communication circuit can request bus BUS-A  
25 during the second-request clock cycle on select lines SEL-B0 and SEL-B1, while a 14th communication circuit can request bus BUS-A during the second-request clock cycle on select lines SEL-A0 and SEL-A1. Thus, in this example, 12 clock cycles after the request period for bus BUS-A began, each of 24 communications circuits has had a one clock

cycle opportunity to request control of bus BUS-A over select lines SEL-A0, SEL-A1, SEL-B0, and SEL-B1.

When requests for bus BUS-A are received during the second request period, state machine 200 moves to state 220 to define a group of requesting circuits that include the communication circuits 116 that  
5 submitted a bus control request during the second request period. For example, state machine 200 can define a group that includes only the communication circuits 116 that requested control of bus BUS-A during the second request period.

10 Following this, state machine 200 moves to state 222 to grant access to one of the group of requesting communication circuits to transmit a data cell in the next transmission period on bus BUS-A. As above, access is granted by outputting a grant to the requesting communication circuit 116 over the control lines SEL-A0 and SEL-A1.  
15 States 220 and 222 can be, for example, eight clock periods long. In addition, an error correction code can be transmitted at the same time on the select lines SEL-B0 and SEL-B1 that are not carrying the grant.

Once the grant has been output, state machine 200 moves to state 224 to wait for a predefined period of time. The total time  
20 required to complete states 218-224 can be, for example, 26 clock cycles. After the predefined time has expired, state machine 200 returns to state 210 to repeat the process. In addition, if no requests are received during the first request period, state machine 200 waits until the end of the 26 clock cycle, and then moves from state 210 to  
25 state 218. Similarly, if no requests are received during the second request period, state machine 200 waits until the end of the 26 clock cycle, and then moves from state 218 to state 210.

Thus, state machine 200 moves through states 210-216, which define a first arbitration period, to determine and grant permission to  
30 transmit a data cell during the next transmission period on bus BUS-B,

and then through states 218-224, which define a second arbitration period, to determine and grant permission to transmit a data cell during the next transmission period on bus BUS-A. As a result, state machine 200 provides an alternating series of arbitration periods where control  
5 over bus BUS-B is determined, and then control over bus BUS-A is determined.

FIG. 3 shows a graphical representation that further illustrates prior art state machine 200. As shown in FIG. 3, state machine 200 defines an alternating series of BUS-B and BUS-A arbitration periods 310  
10 and 312, respectively, on control lines SEL-A0, SEL-A1, SEL-B0, and SEL-B1. The BUS-B arbitration period 310 can be implemented with, for example, states 210-216, while the BUS-A arbitration period 312 can be implemented with, for example, states 218-224.

In the FIG. 3 example, following a BUS-B arbitration period 310,  
15 the communication circuit 116 that received the grant begins transmitting a data cell on bus BUS-B one clock cycle before the next BUS-A arbitration period 312, and continues transmitting the data cell for a transmit period. The transmit period can be, for example, 52 clock cycles long.

20 Similarly, following a BUS-A arbitration period 312, the communication circuit 116 that received the grant begins transmitting a data cell on bus BUS-A one clock cycle before the next BUS-B arbitration period 310, and continues transmitting the data cell for the transmit period.

25 Thus, as shown in FIG. 3, when a communication circuit 116 receives control over one of the two buses, such as BUS-A, the circuit 116 transmits the data cell over the bus during the next two arbitration periods. By utilizing two 26-cycle arbitration periods and one 52-cycle transmit period, a single 52 byte ATM cell can be transmitted.

FIG. 4 shows a flow chart that illustrates a prior art method 400 of granting access to a bus to one of a number of requesting communication circuits. In the present case, each of the requesting communication circuits submitted a bus control request to the bus master during the same request period.

As shown in FIG. 4, method 400, which can be used to implement steps 214 and 222, begins at step 410 by identifying the requesting communication circuits 116 that wish to transmit a high priority data cell. Priority can be divided into different levels, such as high, medium, and low, and assigned to different data cells DS so that a data cell DS with a high priority is preferred over a data cell DS with a medium or low priority, while a data cell DS with a medium priority is preferred over a data cell DS with a low priority.

If no communication circuits with a high priority data cell requested control, method 400 moves to step 412 to identify the requesting communication circuits 116 that wish to transmit a medium priority data cell. If no communication circuits with a medium priority data cell requested control, method 400 moves to step 414 to identify the requesting communication circuits that wish to transmit a low priority data cell.

Once the requesting communication circuits within a priority level have been identified, method 400 moves from either step 410, step 412, or step 414 to step 416 to determine which of the communication circuits within the priority level will receive the grant (permission to transmit during the next transmission period).

When several requesting communication circuits 116 have the same priority level, the requesting communication circuit 116 to receive the grant can be defined by an arbitration, such as a declining round robin. In a declining round robin, the requesting communication circuits 116 circulate within a hierarchical ranking.





addition, a number of the physical addresses are spaced apart (not sequential).

5 The present invention also includes a communication circuit that includes a transmit circuit that transmits information onto a bus, and a receive circuit that receives information from the bus. The communication circuit also includes a memory that has a plurality of physical addresses where each physical address identifies an arbitration period. The plurality of physical addresses identifies a number of arbitration periods.

10 The present invention also includes a logic circuit that is connected to the transmit circuit, the receive circuit, and the memory. If grant information is to be added to the memory, the logic circuit determines a number of desired arbitration periods requested by a communication circuit, and assigns a range of logical addresses that  
15 identify the communication circuit. The number of logical addresses in the range is equal to the number of desired arbitration periods.

A better understanding of the features and advantages of the present invention will be obtained by reference to the following detailed description and accompanying drawings that set forth an illustrative  
20 embodiment in which the principles of the invention are utilized.

#### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a conventional bus-based  
25 communications system 100.

FIG. 2 is a state diagram illustrating a prior art state machine 200 operating as a bus master.

FIG. 3 is a graphical representation further illustrating prior art state machine 200.

FIG. 4 is a flow chart illustrating a prior art method 400 of granting access to a bus to one of a number of requesting communication circuits.

FIG. 5 is a block diagram illustrating an example of a bus-based communications system 500 in accordance with the present invention.

FIG. 6 is a state diagram illustrating an example of a state machine 600 that grants access to a bus to one of a number of requesting communication circuits in accordance with the present invention.

FIG. 7 is a state diagram illustrating an example of a state machine 700 that adds grant information to a memory in accordance with the present invention.

#### DETAILED DESCRIPTION OF THE INVENTION

15

FIG. 5 shows a block diagram that illustrates an example of a bus-based communications system 500 in accordance with the present invention. System 500 is similar to system 100 and, therefore, utilizes the same reference numerals to designate the structures which are common to both systems.

20

As shown in FIG. 5, system 500 differs from system 100 in that system 500 includes a memory 510 that stores grant information that is associated with each of a number of arbitration periods. The grant information, which can be organized as a Priority Table, can be stored in a number of memory devices, including registers and non-volatile memory cells.

25

An example of a Priority Table is shown below. The Priority Table includes a sequential number of physical addresses that represent a corresponding number of rows of data where each row corresponds with an arbitration period. (As described above, an arbitration period can be

30

a 26 clock cycle period where requests are received during a request period and a grant is output.) Thus, a Priority Table that is 512 rows long represents 512 consecutive arbitration periods.

Physical Address	Arbitration Period	Valid Status	Stored Priority	Stored Identity
000000000	1	X	XX	XXXXX
000000001	2	X	XX	XXXXX
000000010	3	X	XX	XXXXX
...	...	...	...	...
111111111	512	X	XX	XXXXX

Priority Table

As shown in the example Priority Table, each row includes a valid status field, a stored identity field, and a stored priority field. The valid status field indicates whether the row of data associated with the arbitration period is valid. The stored identity field identifies a communication circuit 116, while the stored priority field indicates a priority level of a data cell to be transmitted by the communication circuit.

For example, each row of data can include five bits [bits 4:0] of a stored identity field that identify a communication circuit 116, two bits [bits 6:5] of stored priority field that identify one of four priority levels, and a one bit field [bit 7] of valid status that indicates whether the data in the row associated with the arbitration period is valid.

The four priority levels include a high priority level represented by, for example, a 0-0, and a medium priority level represented by, for example, a 0-1. In addition, the four priority levels also include a low

priority level represented by, for example, a 1-0, and an all priority level represented by, for example, a 1-1.

As further shown in FIG. 5, system 500 also differs from system 100 in that system 500 includes a logic circuit 512 that determines which  
5 of a number of requesting communication circuits 116 is next to receive control of a bus. In addition, logic circuit 512 writes the grant information of the communication circuits 116 to various addresses within memory 510.

In addition, system 500 includes a high priority register 514 that  
10 indicates whether the high priority data received by a communication circuit 116 is arbitration, such as round robin, enabled. System 500 also includes a medium priority register 516 that indicates whether the medium priority data received by the communication circuit 116 is arbitration enabled, and a low priority register 518 that indicates  
15 whether the low priority data received by the communication circuit 116 is arbitration enabled.

System 500 can operate the same as state machine 200 in determining a group of requesting circuits, as illustrated in states 210, 212, 218, and 220 shown in FIG. 2. FIG. 6 shows a state diagram that  
20 illustrates an example of a state machine 600 that grants access to a bus to one of a number of requesting communication circuits in accordance with the present invention. State machine 600 can be used to implement states 214 and 222 shown in FIG. 2.

In the present example, the logic circuit 512 of the bus master  
25 has determined the requesting communication circuits that submitted a bus control request to the bus master during the same request period of an arbitration period, such as in states 210, 212, 218, and 220. Method 600 can be executed by the memory 510 and logic circuit 512 of the communication circuit 116 that is the bus master in system 500.

As shown in FIG. 6, state machine 600 begins at state 610 by determining if the grant information that is associated with the arbitration period, such as the information stored in the row of the Priority Table that corresponds with the arbitration period, is valid, such as by checking the valid bit field in the row of the Priority Table. The row of the Priority Table is selected to correspond with the arbitration period. For example, the first row of the table can correspond with the first arbitration period, and the 513<sup>th</sup> arbitration period when a 512 row Priority Table is used.

When the valid bit field indicates that the information is valid, state machine 600 moves to state 612 to determine the identity in the stored identity field that is associated with the arbitration period. For example, the identity can be determined by checking the identity bits in the row of the Priority Table that corresponds with the arbitration period. In addition, state machine 600 also determines in state 612 whether any requesting communication circuit has an identity that matches the identity in the stored identity field.

When a requesting communication circuit has an identity that matches the identity in the stored identity field, state machine 600 moves to state 614 to identify the requesting communication circuit as a matching communication circuit, and determine the priority in the stored priority field that is associated with the arbitration period, such as by checking the priority bits in the row of the Priority Table that corresponds with the arbitration period. In addition, state machine 600 also determines in state 614 whether the priority of the data cell of the matching communication circuit matches the priority in the stored priority field read from the row of the Priority Table.

When the priority of the data cell of the matching communication circuit matches the priority in the stored priority field, state machine 600 moves to state 616 to output a grant to the matching communication

circuit. The grant gives the matching communication circuit 116 permission to transmit during the next transmission period on a bus.

In state 610 when the grant information is invalid, in state 612 when a requesting communication circuit does not have an identity that  
5 matches the identity in the stored identity field, and in state 614 when the matching communication circuit does not have a priority that matches the priority in the stored priority field, state machine 600 moves to state 620 to determine whether any of the requesting communication circuits 116 wishes to transmit a high priority data cell.  
10 In addition, state machine 600 also determines in state 620 which of the high priority communications circuits are also arbitration enabled, such as by checking the high priority registers 514 on the communication circuits 116.

When an arbitration-enabled requesting communication circuit  
15 has a high priority data cell, state machine 600 moves to state 622 to determine which of the arbitration enabled communication circuits within the priority level will receive the grant (permission to transmit during the next transmission period).

When several requesting communication circuits 116 have the  
20 same priority level, the requesting communication circuit 116 to receive the grant can be defined by an arbitration, such as a declining round robin. As noted above, in a declining round robin, the requesting communication circuits 116 circulate within a hierarchical ranking.

After state machine 600 has selected a communication circuit  
25 from the arbitration, state machine 600 moves to state 616 to send a grant to the selected communication circuit. The grant gives the selected communication circuit 116 permission to transmit during the next transmission period on the bus.

If no communication circuits with a high priority data cell  
30 requested control, state machine 600 moves to state 624 to identify the

requesting communication circuits 116 that wish to transmit a medium priority data cell. In addition, state machine 600 also determines in state 624 which of the medium priority communications circuits are also arbitration enabled, such as by checking the medium priority registers  
5 516 on the communication circuits 116.

When an arbitration-enabled requesting communication circuit has a medium priority data cell, state machine 600 moves to state 622 to determine which of the communication circuits within the priority level will receive the grant (permission to transmit during the next  
10 transmission period). As above, the requesting communication circuit 116 to receive the grant can be defined by an arbitration, such as a declining round robin.

If no communication circuits with a medium priority data cell requested control, state machine 600 moves to state 626 to identify the  
15 requesting communication circuits that wish to transmit a low priority data cell. In addition, state machine 600 also determines in state 626 which of the low priority communications circuits are also arbitration enabled, such as by checking the low priority registers 518 on the communication circuits 116.

20 When an arbitration-enabled requesting communication circuit has a low priority data cell, state machine 600 moves to state 622 to determine which of the communication circuits within the priority level will receive the grant (permission to transmit during the next transmission period). As above, the requesting communication circuit  
25 116 to receive the grant can be defined by an arbitration, such as a declining round robin. (State machine 600 can alternately be implemented in software.)

One of the advantages of the Priority Table is that the Priority Table insures that communication circuits with low priority data streams  
30 receive a minimum level of service. For example, to insure that a

communication circuit 116 with a low priority data cell is able to transmit at least once every 512 arbitration cycles, where the Priority Table has 512 rows, the communication circuit 116 can be listed in one row with a low priority level type.

- 5           If a low priority data cell of the communication circuit 116 is to receive only one grant every 512 arbitration periods, then the low priority register 518 on the communication circuit 116 is set to prevent arbitration. If the low priority data cell of the communication circuit 116 is to receive at least one grant, but maybe more than one grant, every  
10 512 arbitration periods, then the low priority register 516 is set to allow arbitration.

Another advantage is that the Priority Table can insure that communication circuits with a high priority data stream receive a known level of service. For example, to insure that a communication circuit 116  
15 with a high priority data stream is able to transmit 150 times every 512 arbitration cycles, where the Priority Table has 512 rows, the communication circuit 116 can be listed in 150 rows with a high priority level type, and arbitration register 514 set to prevent arbitration.

Thus, not only does the Priority Table limit the bandwidth that a  
20 high priority user can have, thereby allowing low priority users to transmit on the bus, the present invention also guarantees the high priority user a predetermined bandwidth that is free from competition from other high priority users. Other levels of service can also be provided by varying the priority level and the enable status.

25           Thus, a communication circuit 116 receives control of a bus based on the contents of the Priority Table. One way to enter data into the Priority Table is to enter the data sequentially. For example, assume a first communication circuit 116 is to receive control of a bus 32 times every 512 arbitration periods.



In this example, the first 32 physical addresses of the Priority Table in memory 510 are assigned to the first communication circuit 116. In other words, the grant information assigned to the first 32 rows of the Priority Table have stored identity fields that are set to identify the first communication circuit 116.

Next, assume a second communication circuit 116 is to receive control of a bus 128 times every 512 arbitration periods, and a third communication circuit 116 is to receive control of a bus 64 times every 512 arbitration periods. In this example, the next 128 physical addresses of the Priority Table in memory 510 are assigned to the second communication circuit 116, while the following 64 physical addresses of the Priority Table in memory 510 are assigned to the third communication circuit 116.

In other words, the grant information assigned to the next 128 rows of the Priority Table have stored identity fields that are set to identify the second communication circuit 116, and the grant information assigned to the following 64 rows of the Priority Table have stored identity fields that are set to identify the third communication circuit 116.

Although this approach is straight forward, this approach also requires that each communication circuit 116 have large amounts of memory to buffer the data cells. For example, assume that 16 communication circuits each want control of a bus 32 times during 512 arbitration periods ( $16 \times 32 = 512$ ), and that the grant information is entered sequentially in memory 510.

In this example, the first 32 arbitration periods grant control of the bus to the first communication circuit 116, which passes 32 sequential cells at the maximum rate of the cell bus interface, such as a rate of 480Mbps. Since first communication circuit 116 outputs such a large burst of data, extensive buffering is required to hold the data.

In the present example, the ideal configuration would be to equally space out the grants to each communication circuit 116 by sequentially granting control of a bus to each of the 16 communication circuits 116. In the ideal configuration, the first communication circuit  
5 116 passes one cell.

Next, the second communication circuit passes one cell, the third communication circuit then passes one cell, and so on. As a result, each communication circuit 116 passes one cell each 16 arbitration periods. Thus, the ideal configuration reduces the size of the data burst from 32  
10 consecutive cells each 512 arbitration periods to one cell each 16 arbitration periods. Therefore, although the same number of cells is transferred during 512 arbitration periods, the size of the memory required to buffer the data can be substantially reduced.

It is often the case, however, that the number of communication  
15 circuits and the number of arbitration periods allocated to each communication circuit vary widely. As a result, it is a non-trivial task to equally space out the bus grants to reduce the buffering requirements of the communication circuits. Further, once the spaced apart memory locations for the bus grant information have been determined,  
20 substantially more operator input time would be required to manually enter the grant information into the proper spaced-apart memory locations.

In the present invention, a scrambling operation is utilized to roughly space apart the bus grants a communication circuit 116 receives  
25 during a period of time, such as 512 arbitration periods, so that the arbitration periods used by each communication circuit can be entered as blocks of logical addresses. As a result, an operator need only enter, for example, that the first communication circuit is to receive 32 arbitration periods, the second communication circuit is to receive 128

arbitration periods, and the third communication circuit is to receive 64 arbitration periods.

FIG. 7 shows a state diagram that illustrates an example of a state machine 700 that adds grant information to a memory in accordance with the present invention. The memory can be implemented with, for example, memory 510, and has a number of physical addresses. The physical addresses correspond with a number of arbitration periods, such as 512, where the number of arbitration periods includes assigned arbitration periods and available arbitration periods. State machine 700 can be implemented in, for example, logic circuit 512.

As shown in FIG. 7, state machine 700 begins at state 710 by determining if grant information is to be added to the memory. When grant information is to be added, state machine 700 moves to step 712 to determine a number of desired arbitration periods requested by a communication circuit.

In the present example, the number of desired arbitration periods is pre-screened to insure that sufficient arbitration periods are available. Alternately, state machine 700 can also compare the number of desired arbitration periods with the number of available arbitration periods in step 712 and, if an insufficient number of arbitration periods are available, state machine 700 can output an error message. The error message can indicate, for example, that an insufficient number of arbitration periods is available.

Once the desired number of arbitration periods has been determined, state machine 700 moves to step 714 to assign a range of sequential logical addresses to the communication circuit. The number of logical addresses in the range, in turn, is equal to the number of desired arbitration periods.

In addition, the memory has a value that represents a next available logical address. When a range of sequential logical addresses is assigned, the first logical address of the range is the next available logical address. In addition, the value that represents the next available  
5 logical address is updated by adding the number of logical addresses in the range to a prior value that represented the next available logical address.

For example, assume that a first communication circuit has 16 arbitration periods and was assigned the range of sequential logical  
10 addresses from 000000000 to 000001111. In this example, the value of the next available logical address is one more or 000010000. When a second communication circuit with 16 arbitration periods is added, a range of logical addresses that runs from 000010000 to 000011111 is assigned to the communication circuit. In this case, the next available  
15 logical address is updated by adding 16 logical addresses to the prior value of 000010000 to obtain an updated value of 000100000.

When a range of logical addresses has been assigned, state machine 700 moves to step 716 to form a number of physical addresses, where a number of the physical addresses are spaced apart, by  
20 changing a number of the logical addresses in the range. Thus, when the range of logical addresses includes more than one logical address, a number of the physical addresses can be spaced apart, e.g., non-sequential, in an address space that represents a predefined time period such as, for example, 512 arbitration periods.

25 Next, state machine 700 moves to step 718 to write grant information for the communication circuit to the memory at the physical addresses. In addition to identifying the communication circuit, the grant information also includes priority information and validity information.

One approach to changing the logical addresses is to use a permutation, such as a reverse permutation. One example of a permutation in accordance with the present invention is shown in the Transformation Table where a logical address is transformed into a physical address.

Logical Address by bit position	8 7654 3210
Physical Address by bit position	0 1234 5678

## Transformation Table

As shown in the Transformation Table, the value of the least significant bit in the physical address is set to have the value of the most significant bit, bit 8, in the logical address. In addition, the value of the next least significant bit in the physical address is set to have the value of the next most significant bit, bit 7, in the logical address. Each bit in the physical address is similarly changed.

The First Example Table shows an example of obtaining physical addresses from three-bit logical addresses by using the permutation illustrated in the Transformation Table.

Com Circuit	Arb Period	Logical Address	Physical Address	Arb Period
A	0	000	000	0
A	1	001	100	4
A	2	010	010	2
A	3	011	110	6
B	4	100	001	1
B	5	101	101	5
C	6	110	011	3
C	7	111	111	7

5

First Example Table

As shown in the First Example Table, communication circuit A has been assigned logical addresses 000 to 011 (which correspond to arbitration periods 0-3), communication circuit B has been assigned  
10 logical addresses 100 to 101 (which correspond with arbitration periods 4-5), and communication circuit C has been assigned logical addresses 110 to 111 (which correspond with arbitration periods 6-7).

However, after each logical address has been processed using the permutation shown in the Transformation Table, the resulting physical  
15 addresses are spaced apart. For example, instead of having arbitration periods 0-3, communication circuit A now has arbitration periods 0, 2, 4, and 6. In addition, instead of having arbitration periods 4 and 5, communication circuit B now has arbitration periods 1 and 5. Further, instead of having arbitration periods 6 and 7, communication circuit C  
20 now has arbitration periods 3 and 7.

Statistical analysis of the permutation shown in the Transformation Table using blocks of grants between 10 and 50

arbitration periods, a 512-row table, a standard deviation  $\sigma = 44.2\%$ , a mean  $\mu = 100\%$ , and a maximum error margin  $\epsilon = 1\%$ , indicates that 34.9% of the time the timing between bus grants to the same communication circuit (where adjacent cells are output by the  
5 communication circuit) will be within 80 to 120% of an ideal timing (where the bus grants are equally spaced-apart).

Thus, one of the advantages of the present invention is that the present invention provides a scrambling operation that spaces apart the grants that a communication circuit 116 receives during a period of time,  
10 such as 512 arbitration periods. As a result, the present invention allows an operator to enter the number of arbitration periods that a communication circuit is to receive in blocks of sequential logical address ranges, while at the same time spacing apart the physical addresses, thereby significantly reducing the buffering required by the  
15 communication circuits 116.

Another example of a permutation in accordance with the present invention is a two-step process where the range of logical addresses is first processed to intermediate addresses by gray encoding, and then the intermediate addresses are processed into physical addresses using  
20 the permutation shown in the Transformation Table.

The Second Example Table shows an example of obtaining the physical addresses from three bit logical address by using the two-step process.

Com Circuit	Arb Period	Logical Address	Inter Address	Physical Address	Arb Period
A	0	000	000	000	0
A	1	001	001	100	4
A	2	010	011	110	6
A	3	011	010	010	2
B	4	100	110	011	3
B	5	101	100	001	1
C	6	110	101	101	5
C	7	111	111	111	7

5

Second Example Table

Like the First Example Table, communication circuit A has been assigned logical addresses 000 to 011 (which correspond to arbitration periods 0-3), communication circuit B has been assigned logical  
10 addresses 100 to 101 (which correspond with arbitration periods 4-5), and communication circuit C has been assigned logical addresses 110 to 111 (which correspond with arbitration periods 6-7).

However, unlike the First Example Table, the logical addresses are first processed by a gray code to form a corresponding number of  
15 intermediate addresses. Following this, the intermediate addresses are processed using the permutation illustrated in the Transformation Table to form the physical addresses.

For example, instead of having arbitration periods 0-3, communication circuit A now has arbitration periods 0, 2, 4, and 6. In  
20 addition, instead of having arbitration periods 4 and 5, communication circuit B now has arbitration periods 1 and 3. Further, instead of having



arbitration periods 6 and 7, communication circuit C now has arbitration periods 5 and 7.

Statistical analysis of the two-step process using the gray code and the permutation shown in the Transformation Table, and blocks of grants between 10 and 50 arbitration periods, a 512-row table, a standard deviation  $\sigma = 40.7\%$ , a mean  $\mu = 100\%$ , and a maximum error margin  $\varepsilon = 1\%$ , indicates that 37.7% of the time the timing between bus grants to the same communication circuit (where adjacent cells are output by the communication circuit) will be within 80 to 120% of an ideal timing (where the bus grants are equally spaced-apart).

It should be understood that the above descriptions are examples of the present invention, and that various alternatives of the invention described herein may be employed in practicing the invention. Thus, it is intended that the following claims define the scope of the invention and that structures and methods within the scope of these claims and their equivalents be covered thereby.